

Urządzenia Peryferyjne Laboratorium

6.12.2023

Grupa nr 7B, Czwartek, godz. 11:00, tydzień nieparzysty

Autorzy:

264488 Adam Czekalski

252560 Karol Szydlak

nr	Treść zadania	data wykonania
4.	Ćwiczenie 6 Skaner płaski. Wprowadzanie informacji do komputera.	23/11/2023

1. Wstęp teoretyczny

Skaner to urządzenie, które służy do przebiegowego przechwytywania obrazu, tekstu, kodu paskowego lub magnetycznego, fal radiowych i przekształcania ich na formę cyfrową. Jest powszechnie używany w biurach, domach, ogólnie w miejscach, w których archiwizuje się dokumenty, tworzy kopie zapasowe i przetwarza obrazy.

- Technologie używane w skanerze
 - CCD (ang. Charge-Coupled Device) – czujnik światła, który przekształca światło zeskanowanego obiektu na sygnał elektryczny, który potem jest przetwarzany na cyfrowy obraz
 - CIS (ang. Contact Image Sensors) – ultracienki czujnik obrazu umieszczony bezpośrednio przy powierzchni którą skanujemy. Składa się z diod LED i czujników światłoczułych. Diody LED emitują światło, które odbija się od dokumentu lub obiektu, po czym czujniki światłoczułe rejestrują to odbite światło, generując obraz cyfrowy
 - LIDE (ang. LED Indirect Exposure) – technologia oparta na diodach elektroluminescencyjnych (LED), które oświetlają dokument lub obraz który jest skanowany. Zaletą tego rozwiązania jest energooszczędność, kompaktowość i wolne nagrzewanie diód
- Podstawowe pojęcia:
 - Kolory cyfrowe: CMYK – model barw, który polega na tym, że kolory są tworzone przez kombinacje barw: Cyjan, Magenta, Żółty, Czarny, które wchłaniają światło
 - Filtry:
 - Filtr kontrastu – pozwala na regulację różnicy między jasnymi a ciemnymi obszarami obrazu
 - Filtr jasności – pozwala dostosować jasność zeskanowanego obrazu
 - Filtr nasycenia barw – pozwala na zmianę intensywności kolorów na zeskanowanym obrazie, w celu uzyskania mniej lub bardziej nasyconych barw
 - Filtr ostrości – pozwala na zwiększenie lub zmniejszenie ostrości krawędzi w zeskanowanym obrazie, co wpływa na jego szczegółowość
 - Balans bieli – proces regulacji kolorów, którego celem jest dostosowanie ich do warunków oświetleniowych, co pomaga w odwzorowaniu rzeczywistych kolorów przechwytywanego obrazu/dokumentu
 - Rozdzielczość skanowania – ilość detali, które skaner jest w stanie zarejestrować podczas procesu skanowania. Najczęściej wyraża się ją w punktach na cal (dpi – ang. dots per inch). Im większa, tym więcej detali jest rejestrowanych
 - De-mozaikowanie – proces przetwarzania obrazów cyfrowych, którego celem jest redukcja efektu mozaiki, czyli efektu widocznego w momencie przybliżenia obrazu/dokumentu, kiedy tracona jest jakość
 - Zoom – w przypadku skanera jest to możliwość ustawienia rozdzielczości skanowania w celu zmiany poziomu powiększenia w trakcie cyfrowego przetwarzania obrazu i pozwala na zwiększenie rozdzielczości skanowania, co może prowadzić do uzyskania lepszej rozdzielczości skanowanego obrazu/dokumentu
- Biblioteki wspierające oprogramowywanie skanerów:

- TWAIN – (ang. Technology Without An Interesting Name)
 - WIA – (ang. Windows Image Acquisition)
 - SANE – (ang. Scanner Access Now Easy)
 - ISIS – (ang. Image and Scanner Interface Specification)
- Formaty zapisu informacji graficznej
 - JPG – stosuje algorytm o stratnej kompresji grafiki rastrowej w celu zmniejszenia rozmiaru pliku obrazka
 - PNG – stosuje algorytm o bezstratnej kompresji grafiki rastrowej
 - TIFF – podobnie jak PNG stosuje algorytm o bezstratnej kompresji, zazwyczaj stosuje się je w przypadkach profesjonalnych ze względu na większy rozmiar
 - BMP – jeden z najstarszych formatów przechowywania obrazów, przechowuje obrazy w formie mapy bitowej. Przechowuje obrazy z maksymalną liczbą szczegółów, nie używając przy tym żadnej kompresji.
 - RLE – prosta technika bezstratnej kompresji danych, która zmniejsza zapotrzebowanie na miejsce poprzez zapisywanie powtarzających się pikseli jako jednostki i ich liczbę powtórzeń

2. Aplikacja

Aplikacja do obsługi skanera płaskiego została napisana w języku C#, korzystając z bibliotek: WIA (Windows Image Acquisition) i System.Drawing oraz z Windows Forms w celu stworzenia interfejsu graficznego.

Poniższy rysunek przedstawia stworzony interfejs graficzny.

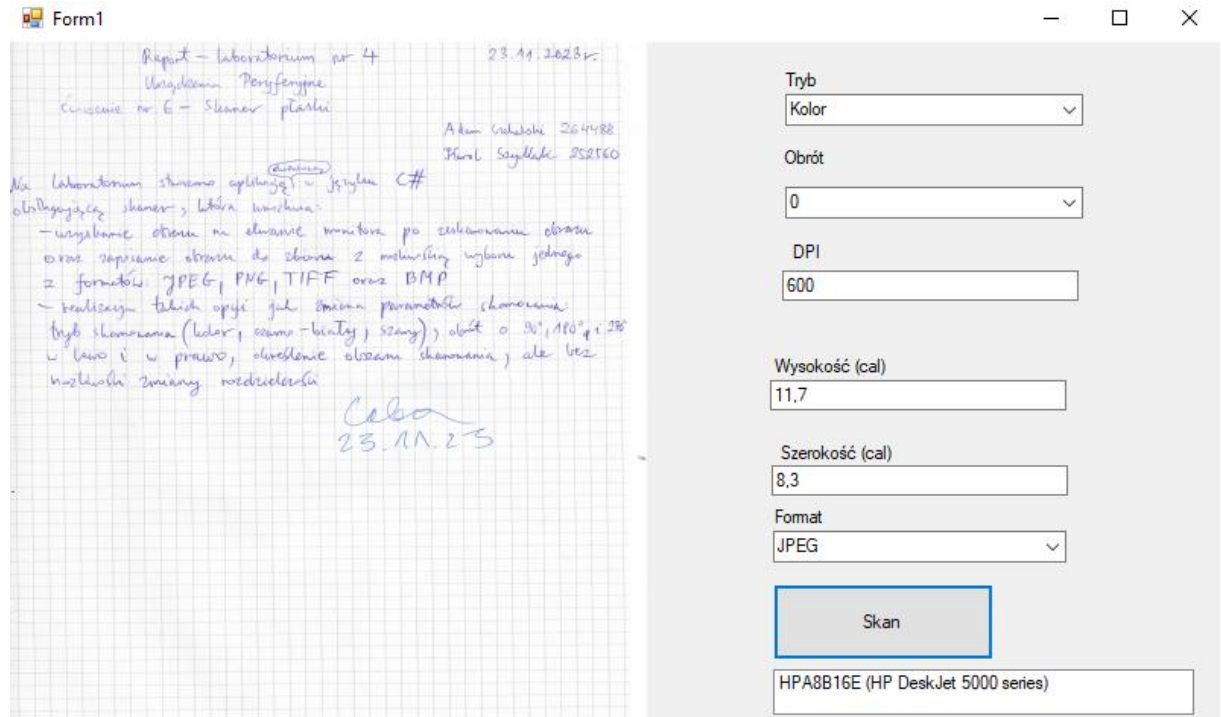
Rysunek 1. Interfejs graficzny aplikacji

Interfejs pozwala na wybór:

- Trybu skanowania(kolor, szary, czarno-biały)
- Obrót zapisanego obrazu o ± 90 , ± 180 , ± 270 stopni
- Wybór obszaru skanowania
- Wybór formatu do zapisu pliku (JPEG, PNG, TIFF, BMP)

2.1. Uzyskiwanie obrazu na ekranie monitora po zeskanowaniu obiektu

Po zeskanowaniu aplikacja wyświetla obraz w oknie graficznym.



Rysunek 2. Zeskanowany obraz

Odpowiedzialne za to są następujące linie kodu:

```
pictureBox1.ImageLocation = path;
pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
```

Rysunek 3. Wyświetlenie zeskanowanego obrazu.

2.2. Zapisanie obrazu do zbioru z możliwością wyboru jednego z kilku podanych formatów
Aplikacja umożliwia wybór formatu zapisu do zbioru z kilku wybranych
(JPEG, PNG, TIFF, BMP). Poniżej przedstawiono odpowiedzialny za to kod.

```
84
85     string format_C = comboBox2.Text;
86     string format = "";
87     string format_end = "";
88     ImageFormat format1 = ImageFormat.Jpeg;
89     //wybór formatu zapisu
90     if (format_C == "JPEG")
91     {
92         format = "{B96B3CAE-0728-11D3-9D7B-0000F81EF32E}"; //JPEG
93         format_end = "jpeg";
94         format1 = ImageFormat.Jpeg;
95     }
96     else if (format_C == "PNG")
97     {
98         format = "{B96B3CAF-0728-11D3-9D7B-0000F81EF32E}"; //PNG
99         format_end = "png";
100        format1 = ImageFormat.Png;
101    }
102    else if (format_C == "TIFF")
103    {
104        format = "{B96B3CB1-0728-11D3-9D7B-0000F81EF32E}"; //TIFF
105        format_end = "tiff";
106        format1 = ImageFormat.Tiff;
107    }
108    else if (format_C == "BMP")
109    {
110        format = "{B96B3CAB-0728-11D3-9D7B-0000F81EF32E}"; //BMP
111        format_end = "bmp";
112        format1 = ImageFormat.Bmp;
113    }
114
115    imageFile = (ImageFile)scannerItem.Transfer(format);
116
117    string path = "C:\\Users\\karol\\Desktop\\testy\\test." + format_end; //ścieżka do zapisu
118
119    if (File.Exists(path))
120    {
121        File.Delete(path);
122    }
123    imageFile.SaveFile(path);
```

Rysunek 4. Wybór formatu zapisu

Dostosowanie formatu zapisu odbywa się poprzez odczytanie wybranej wartości z okna graficznego i za jej pomocą wybranie odpowiedniego identyfikatora, zapisanego w formacie GUID (*globally unique identifier*).

2.3. Zmiana rozdzielczości, obszaru oraz trybu skanowania (koloru)

Zmiana parametrów skanowania odbywa się przy użyciu metody *AdjustScannerSettings*, wygląda ona następująco.

```
155 private static void AdjustScannerSettings(IItem scannerItem, int scanResolutionDPI, int scanStartLeftPixel,
156     int scanStartTopPixel, int scanWidthPixels, int scanHeightPixels, int brightness, int contrast, int color_mode, int rotate)
157 {
158     const string WIA_SCAN_COLOR_MODE = "6146"; // tryb skanowania
159     const string WIA_HORIZONTAL_SCAN_RESOLUTION_DPI = "6147"; //dpi poziome
160     const string WIA_VERTICAL_SCAN_RESOLUTION_DPI = "6148"; //dpi pionowe
161     const string WIA_HORIZONTAL_SCAN_START_PIXEL = "6149"; //początek obszaru skanowania na osi X
162     const string WIA_VERTICAL_SCAN_START_PIXEL = "6150"; //początek obszaru skanowania na osi Y
163     const string WIA_HORIZONTAL_SCAN_SIZE_PIXELS = "6151"; //koniec obszaru skanowania na osi X
164     const string WIA_VERTICAL_SCAN_SIZE_PIXELS = "6152"; //koniec obszaru skanowania na osi Y
165     const string WIA_SCAN_BRIGHTNESS_PERCENTS = "6154"; //jasność
166     const string WIA_SCAN_CONTRAST_PERCENTS = "6155"; //kontrast
167
168
169     SetWIAProperty(scannerItem.Properties, WIA_HORIZONTAL_SCAN_RESOLUTION_DPI, scanResolutionDPI);
170     SetWIAProperty(scannerItem.Properties, WIA_VERTICAL_SCAN_RESOLUTION_DPI, scanResolutionDPI);
171     SetWIAProperty(scannerItem.Properties, WIA_HORIZONTAL_SCAN_START_PIXEL, scanStartLeftPixel);
172     SetWIAProperty(scannerItem.Properties, WIA_VERTICAL_SCAN_START_PIXEL, scanStartTopPixel);
173     SetWIAProperty(scannerItem.Properties, WIA_HORIZONTAL_SCAN_SIZE_PIXELS, scanWidthPixels);
174     SetWIAProperty(scannerItem.Properties, WIA_VERTICAL_SCAN_SIZE_PIXELS, scanHeightPixels);
175     SetWIAProperty(scannerItem.Properties, WIA_SCAN_BRIGHTNESS_PERCENTS, brightness);
176     SetWIAProperty(scannerItem.Properties, WIA_SCAN_CONTRAST_PERCENTS, contrast);
177     SetWIAProperty(scannerItem.Properties, WIA_SCAN_COLOR_MODE, color_mode);
178
179
180
181 }
182
183 Odwołania: 9
184 private static void SetWIAProperty(IProperties properties, object propName, object propValue)
185 {
186     Property prop = properties.get_Item(ref propName);
187     prop.set_Value(ref propValue);
188 }
```

Rysunek 5. Zmiana parametrów skanowania

Odpowiednie parametry skanowania ustawiane są poprzez podanie identyfikatora parametru oraz jego pożądanej wartości w odpowiednim zakresie. Powyższa metoda ustawia tryb skanowania, rozdzielczość, jasność, kontrast oraz obszar skanowania.

2.4. Obracanie obrazu o 90, 180, 270 stopni w obie strony

Za obrót obrazu odpowiada następująca funkcja.

```
122     imageFile.SaveFile(path);
123     Bitmap bt = (Bitmap)Bitmap.FromFile(path);
124     //obrócenie zeskanowanego obrazu
125     if (rotate == 90 || rotate == -270)
126     {
127         bt.RotateFlip(RotateFlipType.Rotate90FlipNone);
128         bt.Save(path);
129     }
130     else if (rotate == 180 || rotate == -180)
131     {
132         bt.RotateFlip(RotateFlipType.Rotate180FlipNone);
133         bt.Save(path);
134     }
135     else if (rotate == 270 || rotate == -90)
136     {
137         bt.RotateFlip(RotateFlipType.Rotate270FlipNone);
138         bt.Save(path);
139     }
```

Rysunek 6. Obrót obrazu

Zeskanowany obraz jest wczytywany jako bitmapa, po czym ponownie zapisywany.

3. Uwagi

W trakcie laboratorium wystąpił problem związany z wyborem rozdzielczości. Wynikał on z wyboru niepoprawnych identyfikatorów parametrów skanowania, błąd został naprawiony.

4. Raport

Raport - laboratorium nr 4	23.11.2023r.
Urządzenia Peryferyjne	
Ćwiczenie nr 6 - Skaner płaski	
	Adam Grelowski 264488
	Harold Szylak 252560
Na laboratorium stworzono aplikację ^{desktopową} w języku C# obsługującą skaner, która umożliwia:	
- wysłanie obrazu na ekranie monitora po zeskanowaniu obrazu oraz zapisanie obrazu do zbioru z możliwością wyboru jednego z formatów: JPEG, PNG, TIFF oraz BMP	
- realizację takich operacji jak zmiana parametrów skanowania: tryb skanowania (kolor, czarno-biały, szary), obrót o 90° , 180° i 270° w lewo i w prawo, obcięcie obrazu skanowania, ale bez możliwości zmiany rozdzielczości	
Coba	
23.11.23	

Rysunek 7. Raport z laboratorium

5. Źródła

- <https://miroslawzelent.pl/informatyka/skanery-ccd-cis-lide-ocr-twain/>
- <https://pl.wikipedia.org/wiki/CMYK>
- <https://www.zs1wyszkow.edu.pl/skaner.htm>
- https://pl.wikipedia.org/wiki/Balans_bieli
- <https://tylkofotografia.pl/artykuly/co-to-jest-demozaikowanie/>
- <https://pl.wikipedia.org/wiki/TWAIN>
- https://en.wikipedia.org/wiki/Windows_Image_Acquisition
- <https://pl.wikipedia.org/wiki/SANE>
- https://pl.wikipedia.org/wiki/Image_and_Scanner_Interface_Specification
- <https://www.adobe.com/pl/creativecloud/file-types/image/comparison/tiff-vs-png.html>
- <https://www.adobe.com/pl/creativecloud/file-types/image/comparison/jpeg-vs-tiff.html>
- <https://www.adobe.com/pl/creativecloud/file-types/image/comparison/bmp-vs-png.html>
- <https://www.adobe.com/pl/creativecloud/file-types/image/comparison/bmp-vs-jpeg.html>